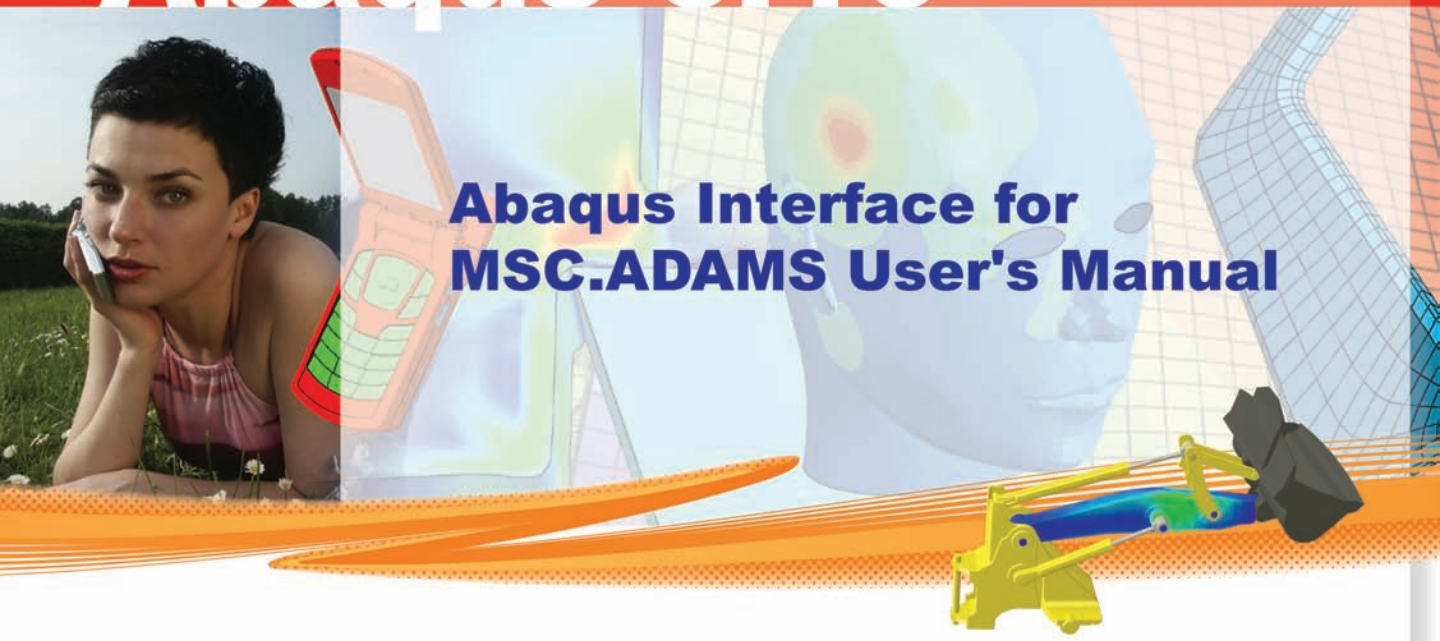
Abaqus 6.10

Abaqus Interface for MSC.ADAMS User's Manual



SIMULIA

Abaqus Interface for MSC.ADAMS

User's Manual

Legal Notices

CAUTION: This documentation is intended for qualified users who will exercise sound engineering judgment and expertise in the use of the Abaqus Software. The Abaqus Software is inherently complex, and the examples and procedures in this documentation are not intended to be exhaustive or to apply to any particular situation. Users are cautioned to satisfy themselves as to the accuracy and results of their analyses.

Dassault Systèmes and its subsidiaries, including Dassault Systèmes Simulia Corp., shall not be responsible for the accuracy or usefulness of any analysis performed using the Abaqus Software or the procedures, examples, or explanations in this documentation. Dassault Systèmes and its subsidiaries shall not be responsible for the consequences of any errors or omissions that may appear in this documentation.

The Abaqus Software is available only under license from Dassault Systèmes or its subsidiary and may be used or reproduced only in accordance with the terms of such license. This documentation is subject to the terms and conditions of either the software license agreement signed by the parties, or, absent such an agreement, the then current software license agreement to which the documentation relates.

This documentation and the software described in this documentation are subject to change without prior notice.

No part of this documentation may be reproduced or distributed in any form without prior written permission of Dassault Systèmes or its subsidiary.

The Abaqus Software is a product of Dassault Systèmes Simulia Corp., Providence, RI, USA.

© Dassault Systèmes, 2010

Abaqus, the 3DS logo, SIMULIA, CATIA, and Unified FEA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of their respective owners. For additional information concerning trademarks, copyrights, and licenses, see the Legal Notices in the Abaqus 6.10 Release Notes and the notices at: http://www.simulia.com/products/products_legal.html.

Locations

SIMULIA Worldwide Headquarters	Rising Sun Mills, 166 Valley Street, Providence, RI 02909–2499, Tel: +1 401 276 4400, Fax: +1 401 276 4408, simulia.support@3ds.com http://www.simulia.com
SIMULIA European Headquarters	Gaetano Martinolaan 95, P. O. Box 1637, 6201 BP Maastricht, The Netherlands, Tel: +31 43 356 6906, Fax: +31 43 356 6908, simulia.europe.info@3ds.com

Technical Support Centers

United States	Fremont, CA, Tel: +1 510 794 5891, simulia.west.support@3ds.com
	West Lafayette, IN, Tel: +1 765 497 1373, simulia.central.support@3ds.com
	Northville, MI, Tel: +1 248 349 4669, simulia.greatlakes.info@3ds.com
	Woodbury, MN, Tel: +1 612 424 9044, simulia.central.support@3ds.com
	Beachwood, OH, Tel: +1 216 378 1070, simulia.erie.info@3ds.com
	West Chester, OH, Tel: +1 513 275 1430, simulia.central.support@3ds.com
	Warwick, RI, Tel: +1 401 739 3637, simulia.east.support@3ds.com
	Lewisville, TX, Tel: +1 972 221 6500, simulia.south.info@3ds.com
Australia	Richmond VIC, Tel: +61 3 9421 2900, simulia.au.support@3ds.com
Austria	Vienna, Tel: +43 1 22 707 200, simulia.at.info@3ds.com
Benelux	Huizen, The Netherlands, Tel: +31 35 52 58 424, simulia.benelux.support@3ds.com
Canada	Toronto, ON, Tel: +1 416 402 2219, simulia.greatlakes.info@3ds.com
China	Beijing, P. R. China, Tel: +8610 6536 2288, simulia.cn.support@3ds.com
	Shanghai, P. R. China, Tel: +8621 3856 8000, simulia.cn.support@3ds.com
Czech & Slovak Republics	Synerma s. r. o., Psáry, Prague-West, Tel: +420 603 145 769, abaqus@synerma.cz
Finland	Vantaa, Tel: +358 46 712 2247, simulia.nordic.info@3ds.com
France	Velizy Villacoublay Cedex, Tel: +33 1 61 62 72 72, simulia.fr.support@3ds.com
Germany	Aachen, Tel: +49 241 474 01 0, simulia.de.info@3ds.com
	Munich, Tel: +49 89 543 48 77 0, simulia.de.info@3ds.com
Greece	3 Dimensional Data Systems, Crete, Tel: +30 2821040012, support@3dds.gr
India	Chennai, Tamil Nadu, Tel: +91 44 43443000, simulia.in.info@3ds.com
Israel	ADCOM, Givataim, Tel: +972 3 7325311, shmulik.keidar@adcomsim.co.il
Italy	Lainate MI, Tel: +39 02 39211211, simulia.ity.info@3ds.com
Japan	Tokyo, Tel: +81 3 5442 6300, simulia.tokyo.support@3ds.com
	Osaka, Tel: +81 6 4803 5020, simulia.osaka.support@3ds.com
	Yokohama-shi, Kanagawa, Tel: +81 45 470 9381, isight.jp.info@3ds.com
Korea	Mapo-Gu, Seoul, Tel: +82 2 785 6707/8, simulia.kr.info@3ds.com
Latin America	Puerto Madero, Buenos Aires, Tel: +54 11 4312 8700, Horacio.Burbridge@3ds.com
Malaysia	WorleyParsons Advanced Analysis, Kuala Lumpur, Tel: +603 2039 9000, abaqus.my@worleyparsons.com
New Zealand	Matrix Applied Computing Ltd., Auckland, Tel: +64 9 623 1223, abaqus-tech@matrix.co.nz
Poland	BudSoft Sp. z o.o., Poznań, Tel: +48 61 8508 466, info@budsoft.com.pl
Russia, Belarus & Ukraine	TESIS Ltd., Moscow, Tel: +7 495 612 44 22, info@tesis.com.ru
Scandinavia	Västerås, Sweden, Tel: +46 21 150870, simulia.nordic.info@3ds.com
Singapore	WorleyParsons Advanced Analysis, Singapore, Tel: +65 6735 8444, abaqus.sg@worleyparsons.com
South Africa	Finite Element Analysis Services (Pty) Ltd., Parklands, Tel: +27 21 556 6462, feas@feas.co.za
Spain & Portugal	Principia Ingenieros Consultores, S.A., Madrid, Tel: +34 91 209 1482, simulia@principia.es
Taiwan	Simutech Solution Corporation, Taipei, R.O.C., Tel: +886 2 2507 9550, lucille@simutech.com.tw
Thailand	WorleyParsons Advanced Analysis, Singapore, Tel: +65 6735 8444, abaqus.sg@worleyparsons.com
Turkey	A-Ztech Ltd., Istanbul, Tel: +90 216 361 8850, info@a-ztech.com.tr
United Kingdom	Warrington, Tel: +44 1 925 830900, simulia.uk.info@3ds.com
	Sevenoaks, Tel: +44 1 732 834930, simulia.uk.info@3ds.com

Complete contact information is available at http://www.simulia.com/locations/locations.html.

Preface

This section lists various resources that are available for help with using Abaqus Unified FEA software.

Support

Both technical engineering support (for problems with creating a model or performing an analysis) and systems support (for installation, licensing, and hardware-related problems) for Abaqus are offered through a network of local support offices. Regional contact information is listed in the front of each Abaqus manual and is accessible from the **Locations** page at www.simulia.com.

SIMULIA Online Support System

The SIMULIA Online Support System (SOSS) provides a knowledge database of SIMULIA Answers. The SIMULIA Answers are solutions to questions that we have had to answer or guidelines on how to use Abaqus, SIMULIA SLM, Isight, and other SIMULIA products. You can also submit new requests for support in the SOSS. All support incidents are tracked in the SOSS. If you are contacting us by means outside the SOSS to discuss an existing support problem and you know the incident number, please mention it so that we can consult the database to see what the latest action has been.

To use the SOSS, you need to register with the system. Visit the **My Support** page at www.simulia.com to register.

Many questions about Abaqus can also be answered by visiting the **Products** page and the **Support** page at www.simulia.com.

Anonymous ftp site

To facilitate data transfer with SIMULIA, an anonymous ftp account is available on the computer ftp.simulia.com. Login as user anonymous, and type your e-mail address as your password. Contact support before placing files on the site.

Training

All offices and representatives offer regularly scheduled public training classes. We also provide training seminars at customer sites. All training classes and seminars include workshops to provide as much practical experience with Abaqus as possible. For a schedule and descriptions of available classes, see www.simulia.com or call your local office or representative.

Feedback

We welcome any suggestions for improvements to Abaqus software, the support program, or documentation. We will ensure that any enhancement requests you make are considered for future releases. If you wish to make a suggestion about the service or products, refer to www.simulia.com. Complaints should be addressed by contacting your local office or through www.simulia.com by visiting the **Quality Assurance** section of the **Support** page.

Contents

1. Introduction

1.1 What information does this manual contain? . . . 1–1
1.2 What is the Abaqus Interface for MSC.ADAMS?. . . . 1–1
1.3 What are the procedures for using the Abaqus Interface for MSC.ADAMS?. . . . 1–1
1.4 What are the contents of the modal neutral file?. . . . 1–2

2. Preparing the Abaqus input file

2.1 The Abaqus substructure model. . . . 2–1
2.2 Setting up the Abaqus model to create a modal neutral file without stress or strain. . . . 2–1
2.3 Setting up the Abaqus model to create a modal neutral file with stress or strain. . . . 2–3
2.4 Supported Abaqus elements . . . 2–4

3. Creating the MSC.ADAMS modal neutral file

3.1 Units. . . . 3–1
3.2 Executing the **adams** command to create a modal neutral file without stress or strain. . . . 3–2
3.3 Executing the **adams** command to create a modal neutral file with stress or strain. . . . 3–4
3.3.1 Creating the second input file. . . . 3–4
3.3.2 Creating the modal neutral file from two results files . . . 3–5
3.4 Translating modes with negative eigenvalues. . . . 3–6
3.5 Diagnosing error messages and problems . . . 3–7

4. Examples

4.1 Example 1: Link modeled with solid elements. . . . 4–1
4.2 Example 2: Link modeled with beam elements . . . 4–5
4.3 Example 3: Tire . . . 4–8

1. Introduction

This chapter provides an overview of the Abaqus Interface for MSC.ADAMS. The following topics are covered:

- "What information does this manual contain?," Section 1.1
- "What is the Abaqus Interface for MSC.ADAMS?," Section 1.2
- "What are the procedures for using the Abaqus Interface for MSC.ADAMS?," Section 1.3
- "What are the contents of the modal neutral file?," Section 1.4

The installation of the Abaqus Interface for MSC.ADAMS is included in the Abaqus product installation. For information on installing Abaqus, see the Abaqus Installation and Licensing Guide.

1.1 What information does this manual contain?

This manual explains how to use the Abaqus Interface for MSC.ADAMS. For general information about using MSC.ADAMS, see the MSC.ADAMS collection of documentation. You might find the following MSC.ADAMS manuals particularly useful:

- *Using ADAMS/Flex*
- *Using ADAMS/View*

1.2 What is the Abaqus Interface for MSC.ADAMS?

The ADAMS/Flex product from MSC.Software Corporation can be used to account for flexibility in a component when performing a dynamic analysis in MSC.ADAMS. ADAMS/Flex relies on a finite element analysis code such as Abaqus to provide the component's flexibility information in a form that is usable by MSC.ADAMS. The Abaqus Interface for MSC.ADAMS can be used to create Abaqus models of MSC.ADAMS components and to convert the Abaqus results into an MSC.ADAMS modal neutral (`.mnf`) file, the format required by ADAMS/Flex.

The Abaqus Interface for MSC.ADAMS 6.10 requires a results file created by Abaqus 6.2 or later. The Abaqus Interface for MSC.ADAMS creates modal neutral files compatible with ADAMS Version 10.1 and later. Modal stress and strain, if present in the modal neutral file, require ADAMS Version 12 or later.

1.3 What are the procedures for using the Abaqus Interface for MSC.ADAMS?

The typical usage of the Abaqus Interface for MSC.ADAMS involves creating a modal neutral file without stress or strain from a single results file and requires one Abaqus analysis and one

Abaqus Interface for MSC.ADAMS step. The following procedure summarizes the typical usage of the Abaqus Interface for MSC.ADAMS:

To use the Abaqus Interface for MSC.ADAMS:

1. Create an Abaqus model for each flexible component of the MSC.ADAMS model. Each component is modeled as an Abaqus substructure.
2. Run the Abaqus analysis.
3. Run the Abaqus Interface for MSC.ADAMS to read the Abaqus results file produced by the analysis and to create the modal neutral (`.mnf`) file for MSC.ADAMS.
4. Read the modal neutral file into MSC.ADAMS. A separate modal neutral file must be created for each flexible part in MSC.ADAMS.

If you want the Abaqus Interface for MSC.ADAMS to translate stress or strain to the modal neutral file, you can modify the general procedure. In the modified procedure the Abaqus Interface for MSC.ADAMS creates a modal neutral file from two results files and requires two Abaqus analyses and two Abaqus Interface for MSC.ADAMS steps. For more information, see "Setting up the Abaqus model to create a modal neutral file with stress or strain," Section 2.3, and "Executing the **adams** command to create a modal neutral file with stress or strain," Section 3.3.

The remaining sections of this manual discuss these procedures in detail.

1.4 What are the contents of the modal neutral file?

The Abaqus Interface for MSC.ADAMS translates data from one or more Abaqus results (`.fil`) files and creates an MSC.ADAMS modal neutral (`.mnf`) file. Depending on the contents of the results files and the translation parameters, the Abaqus Interface for MSC.ADAMS creates a modal neutral file containing the data blocks shown in Table 1–1.

Table 1–1 Modal neutral file contents.

Block Number	Contents	Created by the Abaqus Interface for MSC.ADAMS
1	Version code	Yes
2	Header	Yes
3	Content summary	Yes
4	Nodal coordinates	Yes
5	<Not used>	N/A
6	Global mass properties	Yes

Block Number	Contents	Created by the Abaqus Interface for MSC.ADAMS
7	Eigenvalues	Yes
8	Mode shapes	Yes
9	Nodal masses	Yes
10	Nodal inertias	Yes
11	Units	Yes
12	Generalized stiffness matrix	Yes
13	Generalized mass matrix	Yes
14	Element faces	Yes
15	Generalized damping	No
16	Mode shape transformation	Yes
17	Interface nodes	Yes
18	Modal stress	Optional
19 to 26	Inertia invariants	Yes
27	Modal preload	Yes
28	Modal loads	No
29	Modal strain	Optional

2. Preparing the Abaqus input file

This chapter describes the preparation of an Abaqus input file that will produce the results quantities required by ADAMS/Flex.

2.1 The Abaqus substructure model

The first step in accounting for a component's flexibility in MSC.ADAMS is to model that component as an Abaqus substructure. This process involves creating an Abaqus finite element model of the component. General guidelines for building Abaqus models with substructures are described in "Using substructures," Section 10.1.1 of the Abaqus Analysis User's Manual. The specific requirements for building substructure models that can be exported to MSC.ADAMS are described in the following sections.

The Abaqus Interface for MSC.ADAMS creates modal neutral files that do not contain stress or strain from a single results file that is written by an Abaqus analysis, as described in "Setting up the Abaqus model to create a modal neutral file without stress or strain," Section 2.2. If you want the Abaqus Interface for MSC.ADAMS to translate stress or strain to the modal neutral file, a second Abaqus analysis is required, as discussed in "Setting up the Abaqus model to create a modal neutral file with stress or strain," Section 2.3.

2.2 Setting up the Abaqus model to create a modal neutral file without stress or strain

If you want the Abaqus Interface for MSC.ADAMS to create a modal neutral file without stress or strain, you can use the following template to prepare an input file for the Abaqus analysis:

```
*HEADING
...
**********
*STEP
*FREQUENCY, EIGENSOLVER=...
...
*BOUNDARY
...
*ELEMENT MATRIX OUTPUT, MASS=YES, ELSET=...
*NODE FILE
U,
*END STEP
**********
```

```
*STEP, UNSYMM=NO
*SUBSTRUCTURE GENERATE, TYPE=Z...,
 RECOVERY MATRIX=YES, MASS MATRIX=YES
*RETAINED NODAL DOFS
...
*SELECT EIGENMODES
...
*SUBSTRUCTURE LOAD CASE, NAME=...
*CLOAD
...
*SUBSTRUCTURE MATRIX OUTPUT,
 RECOVERY MATRIX=YES, MASS=YES,
 STIFFNESS=YES, SLOAD=YES
*END STEP
**********
```

The history section of the input file must contain a *FREQUENCY step to calculate the fixed-interface normal modes, followed by a *SUBSTRUCTURE GENERATE step. The *FREQUENCY step may be preceded by any number of steps to apply a desired preload to the model.

Note the following points about the *FREQUENCY step:

- The *FREQUENCY step must apply zero boundary conditions to every degree of freedom that will be retained in the *SUBSTRUCTURE GENERATE step. Other degrees of freedom may be constrained as appropriate.
- This step must write element mass matrices and eigenvectors to the results file.

Note the following points about the *SUBSTRUCTURE GENERATE step:

- The UNSYMM=NO parameter on the *STEP option is optional but recommended. Certain preloading histories (for example, contact with high friction coefficients) may create unsymmetric stiffness matrices. The substructure matrix created after such a preloading history will in all cases be symmetric. However, by default, Abaqus will create and write the full substructure matrix to the results file if the stiffness matrix was unsymmetric earlier in the analysis. Using UNSYMM=NO saves computation time and storage space without affecting accuracy.
- The *SUBSTRUCTURE GENERATE option must include the parameters RECOVERY MATRIX=YES and MASS MATRIX=YES.
- The list of *RETAINED NODAL DOFS must be equal to or a subset of the list of DOFs set to zero in the preceding *FREQUENCY step.
- The list of *SELECT EIGENMODES must be equal to or a subset of the eigenmodes computed in the *FREQUENCY step. If the list is a subset, unused eigenmodes must not be written to the results file.
- The *SUBSTRUCTURE MATRIX OUTPUT option must write the recovery matrix, stiffness matrix, and mass matrix to the results file.

- If the SLOAD=YES parameter is used on the *SUBSTRUCTURE MATRIX OUTPUT option, modal load components corresponding to all internal and external loads acting on the substructure will be translated to modal preloads (block 27) in the modal neutral file.
- The *SUBSTRUCTURE LOAD CASE option is optional. If the option is present, *CLOAD data should duplicate the loading defined in an earlier general step to represent the effect of other parts of the model on the substructure. As noted in the preceding paragraph, if SLOAD=YES is used on the *SUBSTRUCTURE MATRIX OUTPUT option, these external loads (along with any internal loads) will be written to the preload data block of the modal neutral file. The NAME used for the load case is immaterial.

The history section of the Abaqus input file may include general steps preceding the required *FREQUENCY and *SUBSTRUCTURE GENERATE steps. Note the following points about these optional general steps:

- Displacements written to the results file in these steps will be used to update the nodal coordinates written to the modal neutral file. Displacements for later steps will update those of earlier steps. Therefore, if displacements for a subset of nodes have been written for any step, the final step should write displacements for all nodes of the substructure; otherwise, some nodes will be translated with their original coordinates, and others will be translated with their deformed coordinates.
- Loads from these steps may be translated to modal preloads (block 27) in the modal neutral file if SLOAD=YES is used on the *SUBSTRUCTURE MATRIX OUTPUT option. Internal and external loads are treated differently:
  - Internal loads include distributed loads (such as self-weight) and boundary conditions. These loads are considered to be an intrinsic part of the substructure. All *DLOAD data and the reaction forces due to boundary conditions are treated as internal loads.

    The resultant of internal loads may be nonzero. For example, if a gravity load is applied to a structure restrained from rigid body motion, a net force equal to the weight of the body will act on the rest of the MSC.ADAMS model through the retained degrees of freedom.
  - External loads represent the effect of other parts of the model on the substructure. All *CLOAD options are considered to be external loads. To translate these loads to MSC.ADAMS, the *CLOADs in a general step must be replicated as *SUBSTRUCTURE LOAD CASE data in the *SUBSTRUCTURE GENERATE step.

2.3 Setting up the Abaqus model to create a modal neutral file with stress or strain

If you want the Abaqus Interface for MSC.ADAMS to translate stress or strain to the modal neutral file, you must modify the template in the previous section to prepare an input file for the first Abaqus analysis. You must include an output request for stress or strain in the *FREQUENCY step, as shown in the following example:

```
**********
*EL FILE, POSITION=NODES, DIRECTIONS=YES
1,
S,
E,
**********
```

Note the following points about the output request:

- The POSITION=NODES parameter is required.
- The DIRECTIONS=YES parameter is recommended for all models. This parameter is required for models containing shell elements.
- The section point number (**1**, on the line following ***EL FILE**, in this example) is required for models containing shell elements. The section point number will be ignored for solid elements. Stress or strain for only a single section point can be translated to the modal neutral file.
- The output variables stress (**S**), strain (**E**), or both can be written to the results file.

In addition, you must run a second Abaqus analysis to recover stress or strain in the substructure for the static constraint modes. You can create the input file for the second Abaqus analysis using the procedure described in "Creating the second input file," Section 3.3.1. Note the following point about the ∗SUBSTRUCTURE GENERATE step in the first Abaqus analysis:

- If the SORTED=NO parameter is used on the ∗RETAINED NODAL DOFS option and the same node number (or node set) is listed more than once on the data lines, the second input file must be edited so that the corresponding nodes at the usage level appear the same number of times. For more information, see "Ordering of the substructure nodes on the usage level" in "Using substructures," Section 10.1.1 of the Abaqus Analysis User's Manual.

2.4 Supported Abaqus elements

The Abaqus Interface for MSC.ADAMS is designed to support most Abaqus elements that have displacement degrees of freedom at any node. However, there are a few limitations and considerations. Infinite elements (for example, CIN3D8), coupled thermal-structural elements (for example, C3D8T), generalized axisymmetric elements (for example, CGAX3), and frame elements (for example, FRAME3D) are not supported.

3. Creating the MSC.ADAMS modal neutral file

This chapter describes the procedures used to create the MSC.ADAMS modal neutral (**.mnf**) file from the Abaqus results files. The procedure varies depending on whether stress or strain are to be translated to the modal neutral file, as described in "Executing the **adams** command to create a modal neutral file without stress or strain," Section 3.2, and "Executing the **adams** command to create a modal neutral file with stress or strain," Section 3.3.

3.1 Units

The MSC.ADAMS programs require that the user define the units used in the component model, while Abaqus does not. Therefore, during the creation of the modal neutral file the user must declare explicitly the units used in the model. The approach to doing this in the Abaqus Interface for MSC.ADAMS is very similar to the way it is done in the ADAMS/View **Units Settings** dialog box. A predefined units system can be specified by using the **units** option on the Abaqus Interface for MSC.ADAMS execution procedure. Alternatively, the individual length, mass, force, and time units can be specified by using the **length**, **mass**, **force**, and **time** options on the Abaqus Interface for MSC.ADAMS execution procedure. Any individual units that are specified override the corresponding units in the units system. The default units system is **mks**. The valid units systems for the **units** option are listed in Table 3–1.

Table 3–1 Valid units systems.

Units System	Length Units	Mass Units	Force Units	Time Units
mks	meters	kilograms	newtons	seconds
mmks	millimeters	kilograms	newtons	seconds
cgs	centimeters	grams	dyne	seconds
ips	inches	pound-mass	pound-force	seconds

The valid options for each of the **length**, **mass**, **force**, and **time** options are as follows:

Length units

Valid options for the length units are

- **meters**
- **millimeters**, **mm**
- **centimeters**, **cm**
- **kilometers**, **km**
- **inches**, **inch**, **in**

- **feet**, **foot**, **ft**
- **mile**

Mass units

Valid options for the mass units are

- **kilograms**, **kg**
- **megagram**, **tonne**
- **gram**, **g**
- **pound_mass**, **lbm**, **pound**
- **slug**
- **kpound_mass**
- **ounce_mass**

Force units

Valid options for the force units are

- **newtons**, **N**
- **knewton**, **kN**
- **kilogram_force**, **kgf**
- **dyne**
- **ounce_force**
- **pound_force**, **lbf**, **pound**
- **kpound_force**

Time units

Valid options for the time units are

- **seconds**, **sec**
- **milliseconds**, **ms**
- **minutes**, **min**
- **hours**

3.2 Executing the adams command to create a modal neutral file without stress or strain

The **adams** command is used to read the Abaqus results file produced by the multistep Abaqus analysis and to produce an MSC.ADAMS modal neutral (**.mnf**) file. There are several specific requirements on the format of the results file. Creating an input file to satisfy these requirements is described in "Setting up the Abaqus model to create a modal neutral file without stress or strain," Section 2.2.

abaqus adams **job**=*job-name*
[**input**=*input-file*]
[**units**=**mmks** | **mks** | **cgs** | **ips**]
[**length**=*length-units-name*]
[**mass**=*mass-units-name*]
[**time**=*time-units-name*]
[**force**=*force-units-name*]

job

This option specifies the input and output file names to use during results file translation. The *job-name* value is used to construct the default input file name, *job-name*`.fil`. The output modal neutral file is given the name *job-name*`.mnf`.

If this option is omitted from the command line, the user will be prompted for this value.

input

This option specifies the name of the results file if it is different from *job-name*`.fil`.

units

This option specifies the units system for the model. The possible values are **mmks**, **mks**, **cgs**, or **ips**, which correspond to the ADAMS/View options with the same names. The default value is **mks**.

length

This option specifies the length units for the model. If this option is specified, it overrides the length units of the specified units system.

mass

This option specifies the mass units for the model. If this option is specified, it overrides the mass units of the specified units system.

time

This option specifies the time units for the model. If this option is specified, it overrides the time units of the specified units system.

force

This option specifies the force units for the model. If this option is specified, it overrides the force units of the specified units system.

Default values for the units options can be defined in the Abaqus environment file (`abaqus_v6.env`). The default for the **units** option can be defined with the **adams_units_family** parameter. The defaults for the **length**, **mass**, **time**, and **force** options can be defined with the

adams_length_units, **adams_mass_units**, **adams_time_units**, and **adams_force_units** parameters, respectively.

3.3 Executing the adams command to create a modal neutral file with stress or strain

To create a modal neutral file containing stress or strain, the Abaqus Interface for MSC.ADAMS must read data from two results files. You create the first results file according to the requirements described in "Setting up the Abaqus model to create a modal neutral file with stress or strain," Section 2.3. Next, you use the **adams** command to create an input file for the second Abaqus analysis as described in "Creating the second input file," Section 3.3.1. You then run the second Abaqus analysis, which writes the second results file. Finally, you use the **adams** command to create the modal neutral file as described in "Creating the modal neutral file from two results files," Section 3.3.2.

3.3.1 Creating the second input file

To create the input file for the second Abaqus analysis, execute the Abaqus Interface for MSC.ADAMS using the following command:

abaqus adams **job**=*job-name*
[**input**=*input-file*]
[**make_se_recovery**]
[**stress_modes**={**ON** | **OFF**}]
[**strain_modes**={**ON** | **OFF**}]
[**section_point**=*section_point_number*]

job

This option controls the name of the new input file that will be created. The *job-name* value is used to construct the new input file name, *job-name*`_se_recovery.inp`.

If this option is omitted from the command line, the user will be prompted for this value.

input

This option specifies the name of the results file from the first Abaqus analysis, if it is different from *job-name*`.fil`.

make_se_recovery

This option specifies that the translator is to create a new Abaqus input file. A modal neutral file will not be created.

stress_modes

This option specifies that the new Abaqus input file will contain commands to write stress to the results file. The possible values are **ON** and **OFF**. The default value is **ON**.

strain_modes

This option specifies that the new Abaqus input file will contain commands to write strain to the results file. The possible values are **ON** and **OFF**. The default value is **OFF**.

section_point

This option specifies the section point at which shell stresses and/or strains will be written to the results file. The default value is 1. This option will be ignored for three-dimensional continuum elements.

3.3.2 Creating the modal neutral file from two results files

After running the two Abaqus analyses, as described above, you create a modal neutral file containing modal stress or strain by executing the Abaqus Interface for MSC.ADAMS using the following command:

abaqus adams **job**=*job-name*
[**input**=*input-file*]
[**se_recovery_job**=*se_recovery_job-name*]
[**units**=**mmks** | **mks** | **cgs** | **ips**]
[**length**=*length-units-name*]
[**mass**=*mass-units-name*]
[**time**=*time-units-name*]
[**force**=*force-units-name*]

job

This option controls the name of the modal neutral file that will be created. The *job-name* value is used to construct the new modal neutral file name, *job-name*`.mnf`.

If this option is omitted from the command line, the user will be prompted for this value.

input

This option specifies the name of the results file from the first Abaqus analysis, if it is different from *job-name*`.fil`.

se_recovery_job

This option specifies the name of the results file from the second Abaqus analysis.

units

This option specifies the units system for the model. The possible values are **mmks**, **mks**, **cgs**, or **ips**, which correspond to the ADAMS/View options with the same names. The default value is **mks**.

length

This option specifies the length units for the model. If this option is specified, it overrides the length units of the specified units system.

mass

This option specifies the mass units for the model. If this option is specified, it overrides the mass units of the specified units system.

time

This option specifies the time units for the model. If this option is specified, it overrides the time units of the specified units system.

force

This option specifies the force units for the model. If this option is specified, it overrides the force units of the specified units system.

3.4 Translating modes with negative eigenvalues

The Abaqus Interface for MSC.ADAMS uses component modal synthesis to combine the fixed-interface normal modes and the substructure recovery vectors into a basis of modal degrees of freedom that will be used for dynamic analysis in MSC.ADAMS. This modal basis spans a space that includes the rigid body response of the substructure. Typically, for a non-prestressed, unrestrained body in three dimensions, one expects to find six rigid body modes with associated zero eigenvalues.

The situation is, in general, different for prestressed models, where an unrestrained structure may have less than six modes with zero eigenvalues. Prestressing may change the expected zeroes into values that are significantly positive or negative, depending on the sign of the prestress.

By default, the Abaqus Interface for MSC.ADAMS deletes modes with negative eigenvalues and reorthogonalizes the reduced basis. If you want to retain modes with negative eigenvalues, define the environment variable.

- On UNIX platforms type the following command:

```
setenv MDI_MNFWRITE_OPTIONS negative_roots_OK
```

- On Windows platforms type the following command:

```
set MDI_MNFWRITE_OPTIONS=negative_roots_OK
```

In this case the Abaqus Interface for MSC.ADAMS will treat modes with negative eigenvalues in the same manner as all other modes.

To determine if a model will have negative eigenvalues when translated by the Abaqus Interface for MSC.ADAMS, you can add a *FREQUENCY step with no boundary conditions to the input file. If this step is added to the run that creates the results file used by the Abaqus Interface for MSC.ADAMS, it must not write anything to the results file.

3.5 Diagnosing error messages and problems

During execution of the **adams** command, the following warning and error messages may be output:

```
WARNING: There are 'N' elements in substructure Z1, but only 'M'
   -=> mass matrices have been processed from the results file.
   -=> Carefully review this discrepancy before proceeding.
```

Explanation: If the number of mass matrices read is zero, verify that there is an *ELEMENT MATRIX OUTPUT, MASS=YES option in the *FREQUENCY step that preceded the *SUBSTRUCTURE GENERATE step.

If *m* is nonzero but less than *n*, the model may be correct. Some Abaqus elements, such as dashpots, do not have mass matrices. The mass of other elements may be neglected if they are not significant in representing the mass of the substructure. Elements with negligible mass are not required in the element set whose mass is written to the results file.

```
WARNING: No fixed-interface normal modes.
   -=> The results file did not contain any modes from
   -=> a *FREQUENCY step. Typically, this step contains:
   -=> *NODE FILE
   -=> U,
```

Explanation: Vibration mode shapes were missing in the results file. A *FREQUENCY step must precede the *SUBSTRUCTURE GENERATE step. Vibration mode shapes from the *FREQUENCY step are written to the results file using the commands:

```
*NODE FILE
U,
```

Typically, all the mode shapes will be written. However, if a subset of the computed modes is retained in the *SUBSTRUCTURE GENERATE step (using the *SELECT EIGENMODES option), the *FREQUENCY step must write only those modes.

```
ERROR: Missing element mass matrices.
   -=> No element mass matrices were found in the results file.
   -=> The input file must contain an entry similar to:
   -=> *ELEMENT MATRIX OUTPUT, MASS=YES, ELSET=elset_name
```

```
   -=> This entry must be in the *FREQUENCY step preceding
   -=> the *SUBSTRUCTURE GENERATE step.
```

Explanation: The *FREQUENCY step must write element mass matrices to the results file using the syntax given in the error message.

```
ERROR: Missing generalized stiffness and mass matrices.
   -=> Verify that the input file defines this substructure
   -=> and contains the following option:
   -=> *SUBSTRUCTURE MATRIX OUTPUT, STIFFNESS=YES, MASS=YES,
   -=> RECOVERY MATRIX=YES [, SLOAD=YES]
```

Explanation: The *SUBSTRUCTURE GENERATE step must write the generalized (reduced) mass and stiffness matrices, as well as the recovery matrices, to the results file. The substructure load vector may also be written using SLOAD=YES.

4. Examples

This chapter contains three example problems. The first two examples model a simple flexible link component and can be used in the tutorials and examples in the MSC.ADAMS document *Using ADAMS/Flex*. The third example is a tire that is prestressed by inflation and contact with the road prior to creating the Abaqus substructure.

4.1 Example 1: Link modeled with solid elements

This example models a simple flexible link component using three-dimensional continuum elements.

To perform the analysis for the link modeled with solid elements:

1. Enter the following commands to extract the input files from the compressed archive files provided with the Abaqus release:

```
abaqus fetch job=adams_ex1
abaqus fetch job=adams_ex1_nodes
abaqus fetch job=adams_ex1_elements
```

2. Enter the following command to execute the Abaqus analysis:

```
abaqus job=adams_ex1
```

3. Enter the following command to execute the Abaqus Interface for MSC.ADAMS and translate the results file generated in the Abaqus analysis to a modal neutral file for use with ADAMS/Flex:

```
abaqus adams job=adams_ex1
```

The solid element link model used in the MSC.ADAMS four-bar linkage model is shown in Figure 4–1. The link is modeled with 642 C3D10 tetrahedral solid elements (1368 nodes).

Because the solid elements have only displacement degrees of freedom at their nodes, multi-point constraints are used to provide a connection to the other components in the MSC.ADAMS model. Two nodes are added along the centerline of the beam at the centers of the hinge holes. The C3D10 nodes that lie on the faces of the hinge holes are connected to the extra nodes with BEAM-type multi-point constraints, allowing the nodes to transmit both forces and moments between the link and other MSC.ADAMS components.

The options used to define the single substructure are those described in “The Abaqus substructure model,” Section 2.1. Twenty fixed-interface vibration modes are computed to represent the dynamic behavior of the link.

MSC.ADAMS uses the fixed-interface vibration modes and the constraint modes to characterize the flexibility of the link. The eight lowest fixed-interface vibration frequencies computed by Abaqus are shown in Table 4–1. These frequencies are reported in the **adams_ex1.dat** file.

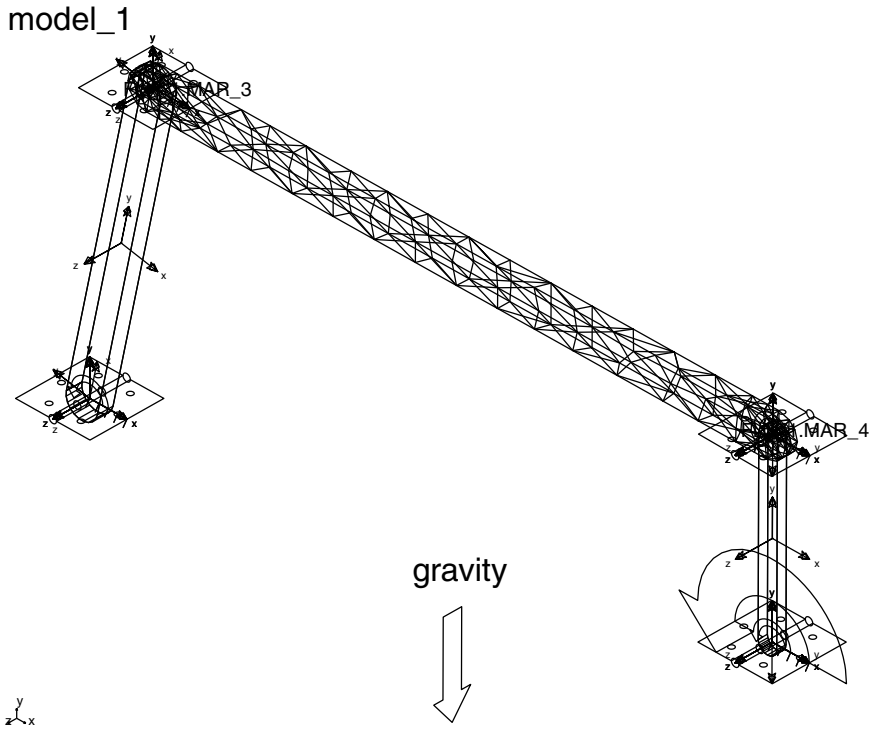


Figure 4–1 Solid link model.

Table 4–1 Fixed-interface vibration frequencies for the solid link model.

Frequency, Hz
206
391
570
1124
1228
1817
1879
2541

The Abaqus Interface for MSC.ADAMS combines these fixed-interface modes with the static constraint modes to compute an equivalent modal basis to be used by ADAMS/Flex. The first six frequencies of this equivalent basis are approximately zero. The next eight frequencies for the unconstrained model are shown in Table 4–2. These frequencies are written to the screen when executing the Abaqus Interface for MSC.ADAMS.

Table 4–2 Nonzero frequencies for the solid link model that are used by ADAMS/Flex.

Frequency, Hz
194
535
574
1055
1551
1762
1801
2653

The Abaqus input file for the solid model, **adams_ex1.inp**, is shown below.

```
*HEADING
Link modeled with C3D10 solid elements
**  ----------------------------------------------
**
**                NODE DEFINITION
**
*NODE,input=adams_ex1_nodes.inp
**
**
*NSET, NSET=LEFTCYL
8,9,17,18,70,71,72,73,125,126,127,128,134,135,207,
229,230,278,309,310,311,312,313,314,373,374,375,376,
377,378,389,390,391,392,498,533,534,535,546,565,566,
677,688,734,1058,1059,1073,1085,1114,1115,1311,1312,
1325,1335,1356,1357
**
*NSET, NSET=RIGHTCYL
6,7,15,16,66,67,68,69,121,122,123,124,136,137,231,
```

```
232,303,304,305,306,307,308,367,368,369,370,371,372,
393,394,395,396,479,480,481,487,488,506,635,654,957,
958,976,977,1004,1026,1219,1220,1234,1235,1257,1287
**
*MPC
BEAM,LEFTCYL,10000
BEAM,RIGHTCYL,20000
**  ----------------------------------------------
**
**               ELEMENT DEFINITION
**
*ELEMENT,TYPE=C3D10,ELSET=PROP1,INPUT=adams_ex1_elements.inp
**
**
**  ----------------------------------------------
**
**               ELEMENT PROPERTY DEFINITION
**
*SOLID SECTION,ELSET=PROP1,MATERIAL=STEEL
**
**  ----------------------------------------------
**
**               MATERIAL DEFINITION
**
*MATERIAL,NAME=STEEL
*ELASTIC
 2.069999944E+11, 3.000000119E-01,
*DENSITY
 7.800000000E+03,
**
*NSET,NSET=RETNODES
10000,20000
**
**  ----------------------------------------------
**
*STEP
*FREQUENCY,EIGENSOLVER=LANCZOS
20,
*BOUNDARY
RETNODES, 1,6
*ELEMENT MATRIX OUTPUT, MASS=YES, ELSET=PROP1
```

```
*NODE FILE
U
*END STEP
**
**  ----------------------------------------------
**
**                SUBSTRUCTURE GENERATION
**
*STEP
*SUBSTRUCTURE GENERATE, TYPE=Z1, RECOVERY MATRIX=YES,
 MASS MATRIX=YES, OVERWRITE
*RETAINED NODAL DOFS, SORTED=NO
RETNODES, 1,6
*SELECT EIGENMODES,generate
1,20
*SUBSTRUCTURE MATRIX OUTPUT, STIFFNESS=YES, MASS=YES,
 RECOVERY=YES
*END STEP
```

4.2 Example 2: Link modeled with beam elements

This example models a simple flexible link component using three-dimensional beam elements.

To perform the analysis for the link modeled with beam elements:

1. Enter the following command to extract the input files from the compressed archive files provided with the Abaqus release:

   ```
   abaqus fetch job=adams_ex2
   ```

2. Enter the following command to execute the Abaqus analysis:

   ```
   abaqus job=adams_ex2
   ```

3. Enter the following command to execute the Abaqus Interface for MSC.ADAMS and translate the results file generated in the Abaqus analysis to a modal neutral file for use with ADAMS/Flex:

   ```
   abaqus adams job=adams_ex2
   ```

The primary difference between the beam model and the solid model is that the beam model uses only 10 B31 elements (11 nodes). Because the beam elements have both displacement and rotational degrees of freedom at their nodes, no multi-point constraints are needed to connect the link to other MSC.ADAMS components. The rest of the model is essentially identical to the solid model of the link.

The first eight nonzero frequencies for the unconstrained model are shown in Table 4–3.

Table 4–3 Nonzero frequencies for the beam link model that are used by ADAMS/Flex.

Frequency, Hz
205
555
610
1070
1618
1742
1775
2568

These frequencies are close to those of the solid model of the link. Although the computational cost in Abaqus is much less for this model than for the solid model, the computational costs in MSC.ADAMS for the two models are very similar because both models have 32 modes (12 constraint modes and 20 fixed-interface vibration modes).

The Abaqus input file for the beam model, **adams_ex2.inp**, is shown below.

```
*HEADING
Link modeled with B31 beam elements
**  ---------------------------------------------
**
**               NODE DEFINITION
**
*NODE, nset=nall
**
         1,0.000000000E+00,0.000000000E+00,0.000000000E+00
         2,5.000000000E-02,0.000000000E+00,0.000000000E+00
         3,1.000000000E-01,0.000000000E+00,0.000000000E+00
         4,1.500000000E-01,0.000000000E+00,0.000000000E+00
         5,2.000000000E-01,0.000000000E+00,0.000000000E+00
         6,2.500000000E-01,0.000000000E+00,0.000000000E+00
         7,3.000000000E-01,0.000000000E+00,0.000000000E+00
         8,3.500000000E-01,0.000000000E+00,0.000000000E+00
         9,4.000000000E-01,0.000000000E+00,0.000000000E+00
        10,4.500000000E-01,0.000000000E+00,0.000000000E+00
        11,5.000000000E-01,0.000000000E+00,0.000000000E+00
**
```

```
**  ------------------------------------------------
**
**                ELEMENT DEFINITION
**
*ELEMENT,TYPE=B31
          1,1,2
          2,2,3
          3,3,4
          4,4,5
          5,5,6
          6,6,7
          7,7,8
          8,8,9
          9,9,10
         10,10,11
**
**  ------------------------------------------------
**
**                ELEMENT PROPERTY DEFINITION
**
*ELSET,ELSET=PROP1
1,2,3,4,5,6,7,8,9,10
**
*BEAM SECTION,ELSET=PROP1,SECTION=RECT,MATERIAL=STEEL,TEMP=GRAD
3.000E-02,1.000E-02
0.000E+00,0.000E+00,-1.000E+00
,
**
**  ------------------------------------------------
**
**                MATERIAL DEFINITION
**
*MATERIAL,NAME=STEEL
*ELASTIC
 2.069999944E+11, 3.000000119E-01,
*DENSITY
 7.800000000E+03,
**
*NSET,NSET=RETNODES
1, 11
**
```

```
**  ------------------------------------------------
**
*STEP
*FREQUENCY,EIGENSOLVER=LANCZOS
20,
*BOUNDARY
RETNODES, 1,6
*ELEMENT MATRIX OUTPUT, MASS=YES, ELSET=PROP1
*NODE FILE
U
*END STEP
**
**  ------------------------------------------------
**
**                SUBSTRUCTURE GENERATION
**
*STEP
*SUBSTRUCTURE GENERATE, TYPE=Z1, RECOVERY MATRIX=YES,
 MASS MATRIX=YES, OVERWRITE
*RETAINED NODAL DOFS, SORTED=NO
RETNODES, 1,6
*SELECT EIGENMODES,generate
1,20
*SUBSTRUCTURE MATRIX OUTPUT, STIFFNESS=YES, MASS=YES,
 RECOVERY=YES
*END STEP
```

4.3 Example 3: Tire

This example models a tire. The substructure is created after solving a highly nonlinear prestress problem to account for inflating the tire and giving it a footprint due to contact with the road.

To perform the analysis for the tire:

1. Enter the following commands to extract the input files from the compressed archive files provided with the Abaqus release:

```
abaqus fetch job=adams_ex3A
abaqus fetch job=adams_ex3A_nodes
abaqus fetch job=adams_ex3B
abaqus fetch job=adams_ex3C
```

2. You must perform three Abaqus analyses.
   a. Enter the following command to solve an axisymmetric model for the tire inflation:

      ```
      abaqus job=adams_ex3A
      ```

   b. Enter the following command to create the three-dimensional model of the tire from the axisymmetric model and its results and to calculate the footprint of the tire in contact with the road:

      ```
      abaqus job=adams_ex3B oldjob=adams_ex3A
      ```

   c. Enter the following command to create the substructure model:

      ```
      abaqus job=adams_ex3C oldjob=adams_ex3B
      ```

3. Enter commands to execute the Abaqus Interface for MSC.ADAMS and to create a modal neutral file for use with ADAMS/Flex.
   - On UNIX platforms enter the following commands:

     ```
     setenv MDI_MNFWRITE_OPTIONS negative_roots_OK
     abaqus adams job=adams_ex3C
     unsetenv MDI_MNFWRITE_OPTIONS
     ```

   - On Windows platforms enter the following commands:

     ```
     set MDI_MNFWRITE_OPTIONS=negative_roots_OK
     abaqus adams job=adams_ex3C
     set MDI_MNFWRITE_OPTIONS=
     ```

This example extends the discussion of the model described in "Symmetric results transfer for a static tire analysis," Section 3.1.1 of the Abaqus Example Problems Manual. The Abaqus analyses of **adams_ex3A** and **adams_ex3B** essentially replicate the inflation and footprint analysis of the tire as described in that section. However, a few modifications have been made to **adams_ex3B** to prepare it for the substructure analysis that follows:

- The rim and hub are modeled as a rigid body, whose reference node is located at the axle. Six degrees of freedom of the reference node will be among the retained degrees of freedom of the substructure.
- The footprint analysis is controlled by applying loads and boundary conditions to this reference node.
- The *MODEL CHANGE, ACTIVATE option is used in the first step of the analysis. This option does not affect the results of that step but is required so that the road-tire contact pair can be removed before creating the substructure in the Abaqus restart analysis of **adams_ex3C**.
- The third step has output requests for CDISP and CSTRESS to determine the tire nodes in contact with the road at the end of the footprint analysis. A subset of these nodes will be among the retained nodes of the substructure.

The tire model in its original and deformed states is shown in Figure 4–2 and Figure 4–3.

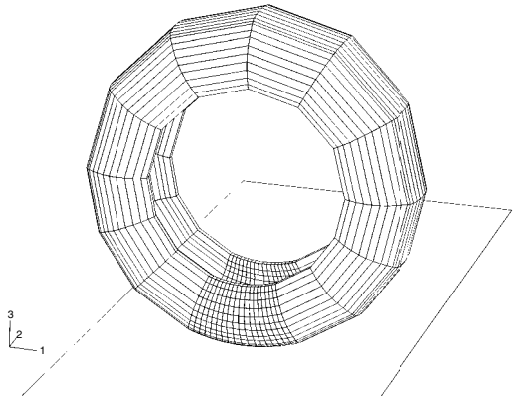


Figure 4–2 Tire model in the original state.

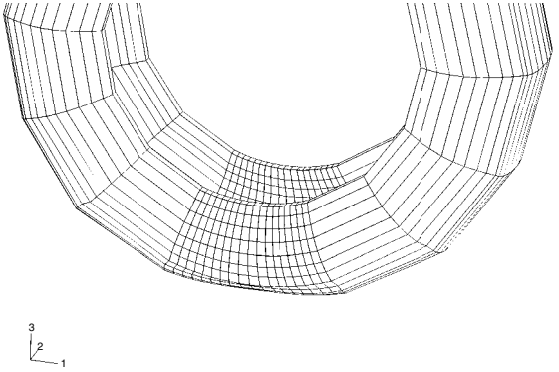


Figure 4–3 Tire model in the deformed state.

The Abaqus analysis of **adams_ex3C** restarts from the inflation and footprint analysis of **adams_ex3B** and consists of the following three steps:

- The tire is isolated from the road. The *MODEL CHANGE, REMOVE, TYPE=CONTACT PAIR option is used to remove the rigid surface representing the road. The mechanics of the solution are unchanged, since the *BOUNDARY, FIXED option is used to specify that the nodes in node set **FOOTPR** have displacements identical to their computed values at the end of the previous step.

One effect of this step is to reformulate the stiffness matrix of the tire without the Lagrange multipliers that were used to enforce the contact constraints; this leads to a more realistic substructure matrix.

This step writes displacements for all nodes to the results file so that deformed nodal coordinates will be written to the results file.

- Twenty normal modes of the tire are computed. This step has boundary conditions to restrain all degrees of freedom that will be retained in the substructure, plus additional restraints to maintain the footprint shape.

  This step writes element mass matrices for all elements and eigenvectors for all modes to the results file. The eight lowest vibration frequencies computed in this step are shown in Table 4–4.

Table 4–4 Fixed-interface vibration frequencies for the prestressed tire.

Frequency, Hz
57
65
70
83
94
99
108
118

To compute the modes and frequencies for the unrestrained, prestressed tire, remove all boundary conditions and run a separate analysis. The eight lowest eigenvalues for this analysis are shown in Table 4–5. The prestress has eliminated two of the zero eigenvalues that would be expected in an unstressed free vibration calculation. These eigenvalues are significantly negative; hence, their retention in the modal neutral file is optional and is controlled by an environment variable, as discussed in "Translating modes with negative eigenvalues," Section 3.4.

- The substructure is created. The list of retained nodal degrees of freedom includes six degrees of freedom at the hub and three degrees of freedom at 35 nodes of the footprint. These contribute 111 degrees of freedom to the substructure. In addition, 20 fixed-interface normal modes are retained, so the substructure mass and stiffness matrices have 131 degrees of freedom. Depending on the engineering use of the substructure, you can choose other retained degrees of freedom. You can experiment with retaining a different number of nodes or possibly only the normal component of displacement at some nodes. In addition, the number of fixed-interface normal modes can be varied.

  The *SUBSTRUCTURE MATRIX OUTPUT option uses the optional parameter SLOAD= YES to write the modal load components to the results file. Thus, after translation, the loads

Table 4–5 Eigenvalues computed by Abaqus for the unrestrained prestressed tire, using all DOFs of the FEA model.

Eigenvalue
−3743
−1970
0
0
0
0
3.048E+05
3.208E+05

corresponding to the fraction of vehicle weight that prestressed the tire will be in the modal neutral file used by ADAMS/Flex.

After reorthogonalizing the component modes computed by Abaqus, the Abaqus Interface for MSC.ADAMS reports the eigenvalues and frequencies of the modes it will store in the modal neutral file. As written to the screen during that translation step, the eigenvalues for the first eight modes are shown in Table 4–6.

Table 4–6 Eigenvalues computed by the Abaqus Interface for MSC.ADAMS for the tire, using component modal synthesis with 20 vibration modes and 111 static modes.

Eigenvalue
−3741
−1969
0
0
0
0
3.139E+05
3.289E+05

The Abaqus input files, **`adams_ex3B.inp`** and **`adams_ex3C.inp`**, are shown below.

adams_ex3B.inp

```
*heading
 tire superelement w/ symmetric results transfer
 step 0: generate full 3d model using tiretransfer_axi_full
 step 1: equilibrate results
 step 2: footprint analysis (displacement control)
 step 3: footprint analysis (load control)
 units: kg, m
*preprint,model=yes,history=yes
*node,nset=road
 9999,   0.0,   0.0, -0.02
*symmetric model generation,revolve,element=200,node=200
 0.0, 0.0, 0.0,    0.0, 1.0, 0.0
 0.0, 0.0, 1.0
 90.0, 3
 70.0, 3
 15.0, 7
 10.0, 4
 15.0, 7
 70.0, 3
 90.0, 3
*symmetric results transfer,step=1,inc=4
*elset,elset=foot,gen
 1001, 4801, 200
 1002, 4802, 200
 1003, 4803, 200
 1004, 4804, 200
 1005, 4805, 200
 1007, 4807, 200
 1008, 4808, 200
 1009, 4809, 200
 1010, 4810, 200
 1011, 4811, 200
 1012, 4812, 200
 1014, 4814, 200
*surface,type=cylinder,name=sroad
 0., 0.,-0.31657,   1., 0.,-0.31657
 0., 1.,-0.31657
 start, -0.3, 0.
 line,   0.3, 0.
*rigid body,ref node=9999,analytical surface=sroad
```

```
*surface,name=stread
 foot, s3
*contact pair,interaction=srigid
 stread, sroad
*surface interaction,name=srigid
*friction
 0.0
*elset,elset=sect,generate
 2800, 3200, 1
*nset,nset=sect,generate
 2800, 3400, 1
*nset,nset=foot,elset=foot
*nset,nset=noutp,generate
 1055, 5055, 200
*file format,zero increment
**************************************************
*step,inc=100,nlgeom=yes
 1:  inflation
*static, long term
** 0.25, 1.0
1.,1.,1.
*model change, activate
*restart,write,overlay
*boundary
 rim_ref, 1, 6
*dload
belt,p5, 200.e3
side,p5, 200.e3
*node print,nset=road,freq=100
 u,
 rf,
*el print,freq=0
*node file,nset=foot,freq=100
*output,field,freq=100
*element output
 s,le
*node output,nset=foot
 u,
*contact output, var=preselect
*output,history,freq=1
*node output, nset=road
```

```
 u, rf
*end step
**************************************************
*step,inc=100,nlgeom=yes
 2:  footprint (displacement controlled)
*static, long term
 0.2, 1.0
*restart,write,overlay
*print,contact=yes
*boundary,op=new
 rim_ref,  1, 6
 road, 1, 2
 road, 4, 6
 road, 3, , 0.02
*node print,nset=road,freq=100
 u,
 rf,
*el print,freq=0
*end step
**************************************************
*step,inc=100,nlgeom=yes
 3:  footprint (load controlled)
*static, long term
 1.0, 1.0
*boundary,op=new
 rim_ref,  1, 6
 road, 1, 2
 road, 4, 6
*cload,op=new
 road, 3, 3300.
*contact print
cdisp,cstress
*end step
**************************************************
```

adams_ex3C.inp

```
*heading
 tire superelement w/ symmetric results transfer
 Restart to identify nodes in footprint
 step 4: remove contact constraints
 step 5: extract fixed interface modes
 step 6: generate superelement
 units: kg, m
*preprint,model=yes,history=yes
*restart,read,step=3,write,overlay
*elset,elset=eall
tread,side,belt
**
**************************************************
*nset,nset=footpr,unsorted
**
** This is the list of tire nodes found to be in contact with the
** road at the end of the previous step.
** (These nodes had status CL in the contact print table.)
**
 1850, 1855, 1905, 2045, 2050, 2055, 2100, 2105, 2245, 2250, 2255,
 2300, 2305, 2440, 2445, 2450, 2455, 2495, 2500, 2505, 2640, 2645,
 2650, 2655, 2695, 2700, 2705, 2840, 2845, 2850, 2855, 2895, 2900,
 2905, 3040, 3045, 3050, 3055, 3095, 3100, 3105, 3240, 3245, 3250,
 3255, 3295, 3300, 3305, 3440, 3445, 3450, 3455, 3495, 3500, 3505,
 3640, 3645, 3650, 3655, 3695, 3700, 3705, 3845, 3850, 3855, 3900,
 3905, 4045, 4050, 4055, 4100, 4105, 4250, 4255, 4305
**************************************************
*nset,nset=footpr_retnodes
**
** This is the list of nodes in the above footprint that will be
** retained in the substructure.
**
 1850, 1855, 1905, 2045, 2100, 2440, 2445, 2450, 2455, 2505, 2500,
 2495, 3040, 3045, 3050, 3055, 3105, 3100, 3095, 3640, 3645, 3650,
 3655, 3705, 3700, 3695, 4045, 4050, 4105, 4100, 4250, 4255, 4305,
 2050, 2105
**
**************************************************
*step,inc=1,nlgeom
 4: remove contact constraints
```

```
*static
1.,1.
*boundary,fixed,op=new
 rim_ref,1,6
 footpr,1,3
 road,1,6
*model change, type=contact pair, remove
 stread, sroad
**
** Write displacements for all nodes to the results file.
** (Needed so the MNF contains deformed nodal coordinates)
*node file
U,
*end step
**
**************************************************
*step
 5: extract fixed interface modes
*frequency, eigensolver=lanczos
20,
**
*boundary,op=new
 road, 1, 6
 rim_ref,1,6
 footpr,1,3
**
** Write element mass matrices to the results file.
*element matrix output, mass=yes, elset=eall
**
** Write eigenvectors to the results file.
*node file
U,
*end step
**
**************************************************
*step
 6: generate superelement
*substructure generate, type=z101, overwrite,
 recovery matrix=yes, mass matrix=yes
**
*boundary,op=new
```

```
 road, 1, 6
*retained nodal dofs, sorted=no
rim_ref,1,6
footpr_retnodes,1,3
*select eigenmodes, generate
1,20,1
*substructure matrix output, stiffness=yes, mass=yes,
 sload=yes, recovery matrix=yes
*end step
**
************************************************
```

About SIMULIA

SIMULIA is the Dassault Systèmes brand that delivers a scalable portfolio of Realistic Simulation solutions including the Abaqus product suite for Unified Finite Element Analysis; multiphysics solutions for insight into challenging engineering problems; and lifecycle management solutions for managing simulation data, processes, and intellectual property. By building on established technology, respected quality, and superior customer service, SIMULIA makes realistic simulation an integral business practice that improves product performance, reduces physical prototypes, and drives innovation. Headquartered in Providence, RI, USA, with R&D centers in Providence and in Vélizy, France, SIMULIA provides sales, services, and support through a global network of regional offices and distributors. For more information, visit www.simulia.com.

About Dassault Systèmes

As a world leader in 3D and Product Lifecycle Management (PLM) solutions, Dassault Systèmes brings value to more than 100,000 customers in 80 countries. A pioneer in the 3D software market since 1981, Dassault Systèmes develops and markets PLM application software and services that support industrial processes and provide a 3D vision of the entire lifecycle of products from conception to maintenance to recycling. The Dassault Systèmes portfolio consists of CATIA for designing the virtual product, SolidWorks for 3D mechanical design, DELMIA for virtual production, SIMULIA for virtual testing, ENOVIA for global collaborative lifecycle management, and 3DVIA for online 3D lifelike experiences. Dassault Systèmes' shares are listed on Euronext Paris (#13065, DSY.PA) and Dassault Systèmes' ADRs may be traded on the US Over-The-Counter (OTC) market (DASTY). For more information, visit www.3ds.com.

Abaqus, the 3DS logo, SIMULIA, CATIA, SolidWorks, DELMIA, ENOVIA, 3DVIA, and Unified FEA are trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries. Other company, product, and service names may be trademarks or service marks of their respective owners.

© Dassault Systèmes, 2010

DASSAULT SYSTEMES